動的情報資源に基づくネットワーク管理情報の利用支援に関する研究

平成20年度 修士学位論文

本審査資料

能動的情報資源に基づく
ネットワーク管理情報の利用支援に関する研究

東北大学大学院 情報科学研究科
応用情報科学専攻
博士課程前期2年の課程
情報通信ソフトウェア学講座
木下・阿部研究室

A7IM4001
Bhalla Isha

平成20年2月18日
10:20~10:45
第III会場（204講義室）

目 次

第1章　序論

本研究は，ネットワーク管理支援システム（NMS:Network Management Support System）の管理負担を軽減し，管理者にやさしいNMSを実現することが目的である．第1章では，本研究の背景と目的，本論文の構成について述べる．本章の構成として，はじめにに近年のネットワーク管理の現状について説明し，その問題点を述べる．次に本研究の目的について述べ，おわりに本論文の構成を述べる．

1.1　研究の背景

コンピュータの低価格化やインターネットの普及により，ネットワーク上で多種多様なサービスが提供されるようになった．これに伴い，ネットワークサービス利用者やネットワークサービスそのものの重要性も増してきており，ネットワークサービスを管理するネットワークシステムは大規模化，複雑化の傾向にある．このことから，ネットワークサービスが提供不可能になった場合の損害も大きい．

このようなネットワークシステムを管理・運用するためには，ネットワークの管理者には，高度な経験的知識および煩雑な作業が求められ，これらの作業の際，多様な情報が発生する．そこで，それらの情報を効率的に利用する為の手段として，各種のネットワーク情報解析ツールやネットワーク管理システムなど，情報を抽出・管理するためのツールが提案されている．しかしながら，これらの既存手法は管理対象が特定の機能や層に限定される傾向にあり，収集した情報の中から，必要な情報を抽出する際の管理者の作業負担が大きい．また，ネットワーク管理システムが高度化するにつれ，これを管理者が使いこなすには，複雑な管理スキルが求められる．そのため，ネットワーク管理者は解析ツールや

ネットワーク管理システムを利用しつつも，最終的には，日頃の経験や知識に基づいて，必要な情報の抽出や管理を行っている．また，ネットワーク管理システムの多くは，ネットワーク環境の変化（ネットワーク機器の設定の変更やネットワーク構成の変更）を想定した設計になっておらず，環境に変化があった場合，ネットワーク管理システムが自動的に対応することは困難である．このため，ネットワーク管理者の作業負担は依然として大きい．

1.2　研究の目的

現状のAIR-NMSの情報取得および保持機能は，以下のような設計に基づいて実現されている．

- 情報の取得は，管理者がネットワークドメインごとに手動で取得
- I-AIRの情報資源は，ネットワーク管理者が取得した情報に基づいて，環境ごとに固定的に設計

しかしながら，情報の取得や保持方法はネットワークドメイン環境（提供されているサービス，ネットワークポリシー，ネットワーク構成等）により異なる．そのため，現状のAIR-NMSにおいては，特定のネットワークドメインに適応したAIR-NMSを異なるネットワークドメインに適応する場合，新たに情報を取得・格納することによって，I-AIRの情報資源を用意する必要がある．このことから，AIR-NMSを異なるネットワークドメインに適応する場合に管理者の作業負担が発生するという問題が存在する．

そこで本研究では，NMSの管理負担を軽減し，管理者に優しいNMSの実現を目的とする．具体的には，これまでNMSの管理業務の中で，ネットワークの設定や構成情報に変化があった場合の情報の取得および格納業務に焦点を当て，以下の機能の実現をすることによって，この目的の達成を目指す．

- 自律的・能動的な情報取得支援機能

- 運用・再利用が容易な情報管理機能

1.3　本研究の効果

本研究で提案する手法を導入することによって，ネットワークや機器の設定に変更があった場合の管理者の作業負担軽減が期待できる．これまで，ネットワークや機器の設定に変更があった場合，管理者が手動で機器の設定やネットワークの設定の変更をシステムに反映させた上で，NMS を用いた自動的なネットワークの運用を行っていた．しかし，提案手法を導入することによって，管理者が手動で機器の設定やネットワークの設定の変更をシステムに反映させる際の作業相当部分を，システムで自動化することができ，この際の管理者の作業負担を軽減することが可能になる．

1.4　本論文の構成

本論文では，以下の構成に従って，論ずる．第2章では，能動的情報資源の概念，および本研究の関連研究について述べ，さらに本研究の焦点であるI-AIR の構成要素を持つAIR-NMS の詳細について説明する．第3章では，提案手法を実現するために必要な要件に基づいたI-AIR の詳細設計，エージェントフレームワークを用いた場合のI-AIR の実現方法について述べる．第4章では，第3章で述べた提案手法に基づいて実装した試作システムについて述べ，提案手法の有効性を示すための実験および評価を行う．最後に第5章において結論を述べる．

第2章　能動的情報資源を用いたNMS

本章では，本研究の関連研究について述べる．本章の構成として，まず一般的なNMSおよびエージェントを用いたNMSについて関連研究として述べ，次に，能動的情報資源（AIR）およびAIRの概念をネットワーク管理に適応したAIR-NMSの全体像について述べる．最後に，AIR-NMSにおけるI-AIRおよびK-AIRの機能と役割について述べる．

2.1　一般的なNMS

2.1.1　代表的なNMSの種類

NMSの種類として，代表的なものには次のものがある．

- 集中管理型NMS

  1台または数台のコンピュータを用いて，ネットワーク上の各マシーンに問い合わせを行い，デバイスやパフォーマンスに関する様々な情報を収集し，管理支援する方式である．この方式の特徴として，1台のコンピュータで多くのネットワークの状態を把握できる反面，管理コンピュータに高い性能が要求されたり，管理コンピュータに何らかの問題が起きた場合に管理機能全体に支障を及ぼす可能性がある．

- 分散管理型NMS

  ノードの情報の収集は各ノードに行わせ，加工された情報を一箇所に収集する管理方式である．この方式の特徴として，収集情報の分散化を図ることができ，管理機能の一部がダウンしてもネットワーク管理を継続できる反面，各ノードから情報を収集する際や収集情報を統合する際の負担が大きい．

- エージェント型NMS

  エージェントは，プログラムモジュールと違い，ある事象が起きたときにどのように振舞うかの知識を持つため，利用者がエージェントを制御しなくても自動的に動作（自律性）し，他のエージェントに働きかけることでタスクを完了（能動性）させることができる．このことから，エージェント型NMSは従来の分散処理型NMSの利点に加え，各ノードの情報が自動収集され，他のネットワークに再利用できる利点がある．一方で，例えば，モバイルエージェントの場合，ノードの情報収集巡回経路を事前に設定する必要があり，このような場合，巡回経路の変更への対応が困難である．

2.1.2　関連研究

ネットワークの大規模化や複雑化を受け，これらを管理するには，ネットワークの規模や多様性に応じたNMSが求められている．そこで，分散ネットワーク環境において，ノードにそれぞれモバイルエージェントを配置し，管理を行うエージェント型NMSとしてDOORS[1]が提案されている．この研究では，モバイルエージェントをリポジトリーにおいて管理し，デバイスに極めて近いノードに行くように構成する．リポジトリーでは，クライアントからのデータリクエストの管理・調整を行い，エージェントがデバイスに到着した後，管理デバイスをポーリングやクライアントのリクエストを処理した上で，結果をリポジトリーに返す仕組みになっている．この手法により，エージェントを利用することによって，クライアントのリクエストをより機能的に処理することが可能になる．しかしながら，このシステムでは，情報を効率的に取得することができるが，取得したデータの値をデータベース化して保持している為，データの値が何を表すのかを管理者が把握することは困難であるといえる．

2.2　能動的情報資源を用いたNMS

2.2.1　能動的情報資源

現在，インターネットを代表とする分散環境上には，膨大な量の各種情報資源（文字，記号，画像，図形，音，マルチメディアデータなど種々の対象を電子化したもの）が蓄積されている．このような分散情報資源を活用する為に，一般的には，検索/加工/統合といった処理が情報資源に対して必要であり，この際の手間は，専門的な知識や専門のツールを持たない利用者が，分散情報資源を利用する際の大きな障害となっている．そこで，この問題を解決し，分散情報資源の有効活用を図るための手法として，能動的情報資源（AIR: Active Information Resource）が提案されている [2]．この手法では，情報資源に利用支援知識と利用支援機能からなる利用支援機構を付加することで，エージェントとして各AIRが構成される（図2.1）．

- **利用支援知識**

  情報資源の内容や情報資源を有効に活用するための方法などを表す知識

- **利用支援機能**

  不足している情報を他のAIRに問い合わせて補完する，外部からの要求に対して情報資源から必要な部分を抜き出すなど，実際に情報資源を加工する機能

これにより，情報資源自体が能動性・自律性を持つことになり，これまで利用者が行っていた情報資源の検索や利用にかかわる諸作業の一部を情報資源自体が代行することが可能になる．例えば，利用者から利用要求のメッセージを受けたAIRは，自身が持つ利用支援機能や利用支援知識に基づき，情報資源に関する処理を自動で発動し，他のAIRと協調・連携をすることで利用者の要求に応えることが可能になる．また，図2.1のように複数の情報資源をAIRとしてワークプレイスに配置することによって利用者の要求に複数のAIRが協調・連携して柔軟に代行することが可能になる．

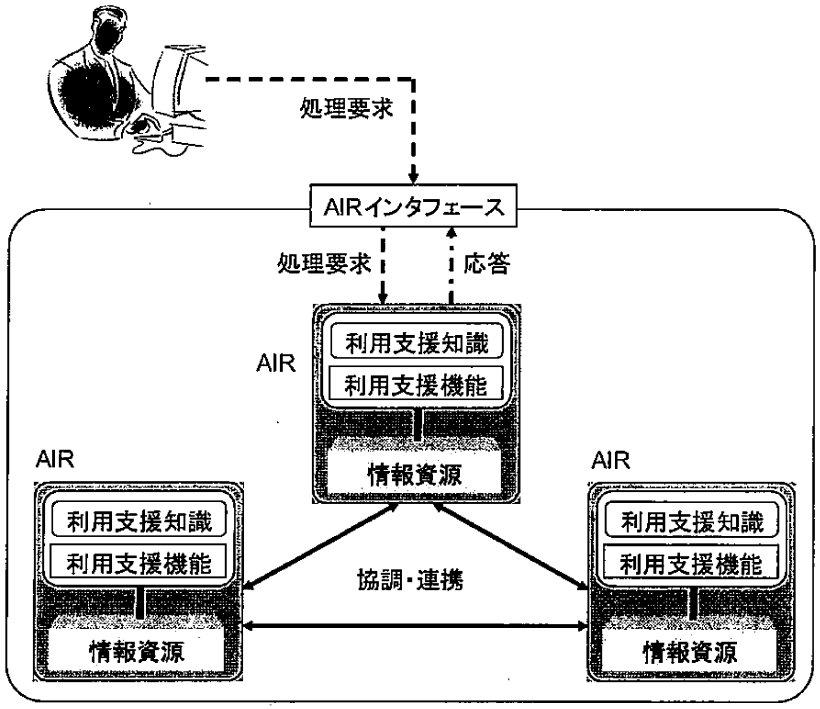

図 2.1: 能動的情報資源の概念図

2.2.2　能動的情報資源を用いたネットワーク管理支援システム (AIR-NMS)

通常，ネットワークの維持・管理は，ネットワークを構成する各機器から取得する状態情報などネットワーク内に分散した情報と，管理者の経験的知識とを用いて行われる．このため，ネットワークが大規模化・複雑化するにつれ，これらの作業を行う管理者の作業負担は増加し，管理者は，ネットワーク管理に関する一般的な知識に加え，管理対象ドメインに関する固有の知識にも精通していることが求められる．能動的情報資源の概念を用いたネットワーク管理支援システム（AIR-NMS: AIR based Network Management Support System，図 2.2）[3] とは，このようなネットワークの状態情報や管理者の経験的知識を分散情報資源とみなして AIR の概念を適応し，ネットワーク管理を支援する仕組みである．ネットワーク上の各情報資源が AIR として協調・連携することによって，次のようなことが可能になり，高度かつ柔軟なネットワーク管理支援と管理者の作業量の軽減が期待できる．

- 管理作業の一部を AIR 側で代行
- ネットワーク上に存在するすべての情報資源を一元的に統合
- 経験的管理知識を用いた柔軟なネットワーク管理
- 経験的管理知識の更新／追加／継承
- 状態情報・構成情報の分散管理

次の節では，AIR-NMS の構成要素とその駆動方式について説明する．

2.2.3　AIR-NMS のアーキテクチャ

AIR-NMS はネットワーク情報（構成情報，状態情報）に AIR の概念を適用した I-AIR (Information AIR）と経験的管理知識に AIR の概念を適用した K-AIR（management Knowledge AIR）の 2 種類の AIR によって構成されている．

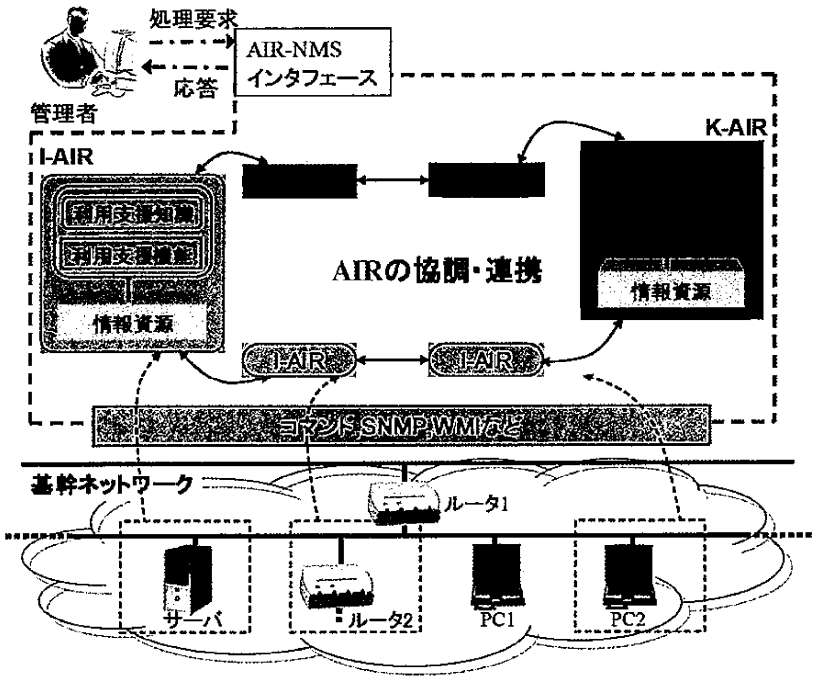

図 2.2: AIR-NMS の流れ

- I-AIR（Information AIR）

  ネットワーク上にある端末の状態情報や構成情報，各種サービスの設定および解析ツールや NMS などから得られる情報を情報資源とする AIR

- K-AIR（management Knowledge AIR）

  テキスト形式の文章で記述された経験的管理知識を情報資源とする AIR

AIR-NMS は分散環境上に配置された AIR が互いに情報を共有することによって，ネットワークの障害への対策支援を行う．具体的な障害対策支援の流れは図 2.3 に示す．まず，ネットワーク上に障害が発生した場合，管理者は AIR のインタフェースに障害現象と障害状況を入力する．AIR のインターフェースは，その入力内容に基づいて K-AIR に障害の調査を依頼する．K-AIR は，依頼を受け自身の持つ管理知識のうち，どの管理知識を用いるか判断する．この際，ネットワーク固有の情報が必要であると判断した場合，この

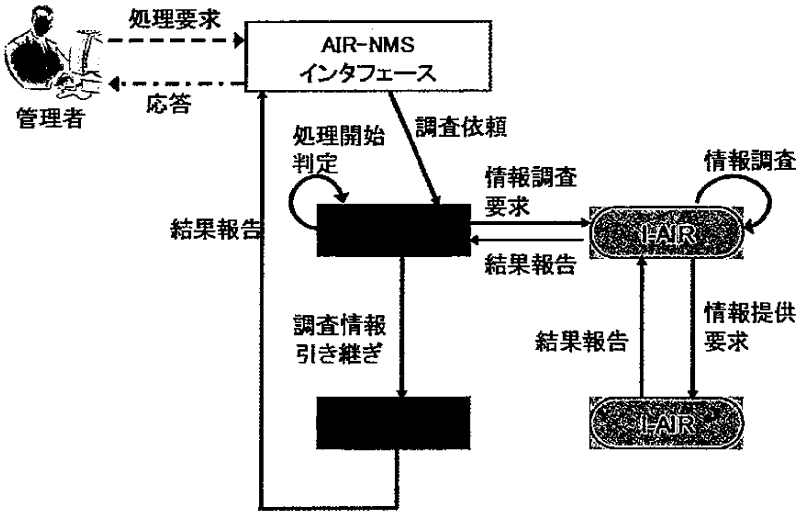

図 2.3: 障害対策支援の流れ

情報に関してネットワーク情報をもつI-AIRに問い合わせることで，障害の原因を特定する．最後に，得られた障害原因をAIRインタフェースを通して管理者に掲示することでシステムは成り立っている．

このように，AIR-NMSでは分散環境上に配置されたAIRが互いに情報を共有することで，ネットワークの障害への対策支援を行っており，これを実現するための具体的なプロセスとして要求ベース駆動とアラームベース駆動の2つの駆動方式が用意されている．

- **要求ベース駆動**

  要求ベース駆動は，AIR-NMSを利用するネットワーク管理者からAIR-NMSインタフェースを通して各AIRに対して情報調査・障害原因特定の依頼を受け，AIR-NMSを駆動させる受動的な駆動方式である．各AIRは自身の持っている情報資源をもとに依頼に対する調査を行い，他のAIRと協調・連携を行うことで最終的な結果を生成してネットワーク管理者に結果を提示する．

- **アラームベース駆動**

  アラームベース駆動はI-AIRが自身の情報資源の変化を検知し，自律的にAIR-

NMSを駆動させる能動的な駆動方式である．例えば，AIR-NMSが管理している端末のログに新しいエラーメッセージが追加された場合には，ログを情報資源とするI-AIRがエラーメッセージの内容から障害を検知して他のAIRに障害内容を通知することで，自動的な障害原因特定を実現する[4].

2.2.4　I-AIRの役割と機能

I-AIRはネットワーク構成情報，ネットワーク上の端末の状態情報，サービスアプリケーションなどの設定，ネットワーク解析ツールやNMSから得られる出力など，ネットワーク上に分散して存在するさまざまな情報（ネットワーク情報）をRDF/XML[5]形式に変換し，情報資源として保持する．そして他のAIRからの要求に応じて自身の持っている情報を適切に加工して提供する役割を持つ.

RDF/XML形式は，情報についてのメタデータの表現方法についての枠組みであり，これを用いることでコンピュータが扱う情報の分類や検索などの自動化・効率化を図ることができることから，近年さまざまな情報がRDF/XML形式によって記述されている．RDF/XML形式による情報資源の表現とAIRの機能を組み合わせることにより，表現形式の違うさまざまなネットワーク情報をI-AIRとして一元的に利用することが可能となり，これまで利用できなかった情報をネットワーク管理に利用することが可能となる．I-AIRの情報資源の例を図2.4に示す.

AIR-NMSにおけるI-AIRの役割は，次の2つである.

- ネットワーク構成や状態の変化に応じた情報資源の更新
- 外部からの要求に対して、適切な情報を提供

例えば，ネットワーク構成情報を情報資源として与えられたI-AIRは，DHCP機能による端末のIPアドレスの変化を検知して情報資源に反映したり，DNSの再起動コマンドを要求された場合にDNSのプロセス名やDNSサーバのIPアドレス，OSを調査し適切

```
            (中略) ・・・
-<NIC>
  <Caption>[00000001] RAS 非同期アダプタ</Caption>
  <ServiceName>AsyncMac</ServiceName>
  <DHCPEnabled>0</DHCPEnabled>
  <IPAddress>IPAddress</IPAddress>
  <DHSHostName>Not Found!!</DHSHostName>
  <DNSServerSeachOrder>Not Found!!</DNSServerSeachOrder>
  <MACAddress>MACAddress</MACAddress>
  <IPSubnet>IPSubnet</IPSubnet>
  <DefaultIPGateway>DefaultIPGateway</DefaultIPGateway>
  </NIC>
- <NIC>
  <Caption>[00000002] WAN ミニポート (L2TP)</Caption>
  <ServiceName>Rasl2tp</ServiceName>
  <DHCPEnabled>0</DHCPEnabled>
  <IPAddress>IPAddress</IPAddress>
  <DHSHostName>Not Found!!</DHSHostName>
  <DNSServerSeachOrder>Not Found!!</DNSServerSeachOrder>
  <MACAddress>MACAddress</MACAddress>
  <IPSubnet>IPSubnet</IPSubnet>
  <DefaultIPGateway>DefaultIPGateway</DefaultIPGateway>
  </NIC>
```

図 2.4: I-AIR の情報資源の例

なコマンドなどを返す．また，端末の状態情報やNMSの出力を情報資源として与えられたI-AIRは，情報資源の内容の変化から障害の発生を検知し，他のAIRに通知する役割を持つ．

これらをAIRの機能として実現するために，各I-AIRは，他のAIRからの情報問い合わせメッセージを解釈し，自身の持つ情報資源を適切に加工して提供したり，ネットワーク情報の更新を監視し情報資源を管理（書き換え／障害を検知）したりするための利用支援知識・利用支援機能を持ち，AIR同士が能動的に協調・連携している．I-AIRの各ステップの詳細は，次のとおりとなる．

1. **調査依頼の通知**

   要求ベース駆動の場合は，管理者がインタフェースを通して入力した障害内容を基にK-AIRが必要な情報を判断し，通知してきた調査依頼メッセージを受信する．アラームベース駆動の場合は，I-AIRが障害を検知し，障害のパターンに応じた調査依頼メッセージをK-AIRに通知する．

2. **処理開始判定**

   調査依頼のメッセージを受信したI-AIRは，メッセージの内容を判定し，自身の持つ知識に不足する情報を補完するために，情報を新たに取得しなおす必要があるのかどうか判断する．そして，必要であると判断した場合，I-AIRは情報取得を開始する．

3. **調査情報の引き継ぎ**

   情報取得が完了すると，情報格納用I-AIRに調査結果を通知し，情報資源に取得結果を格納する．

4. **調査結果の出力**

   K-AIRからの要求に対して，取得した情報を基に応答を返す．

2.2.5　K-AIRの役割と機能

K-AIRは特定のネットワークドメイン内で発生しうる障害への対策を策定する為に必要な作業手順をテキスト形式で記述し，これをを情報資源とするAIRである．AIR-NMSのネットワークサービス障害原因特定においては，1つのK-AIRが調査した調査結果を他のK-AIRが引き継ぎ，引き継いだ情報をもとにより詳細な情報を調査するという協調・連携動作を繰り返すことで最終的に詳細な原因を特定する．ネットワークサービス原因特定の各ステップの詳細は以下のようになる．

1. **調査依頼の通知**

   要求ベース駆動の場合は，管理者がインタフェースを通して障害内容を入力することでK-AIRに対して調査依頼メッセージを通知する．アラームベース駆動の場合は，I-AIRが障害を検知し，障害のパターンに応じた調査依頼メッセージをK-AIRに通知する．調査依頼の段階では，メッセージの内容には発生した障害の内容（例えば,「メールが送受信できない」や「名前解決ができない」）のみが含まれている．

2. **処理開始判定**

   調査依頼のメッセージはAIR-NMS上に存在するすべてのK-AIRに対して通知される．このとき，各AIRがメッセージの内容を判定し，自身の持つ経験的管理知識を用いて障害原因を特定可能かどうかを判断する．そして，障害内容と現在判明している情報から原因特定が可能と判断すると，K-AIRは原因特定のための調査を開始する．

3. **情報の調査**

   K-AIRは自身の持つ経験的管理知識をもとにI-AIRにネットワーク構成情報や端末の状態情報を問い合わせることで，ネットワークの状況を調査する．そしてI-AIRから得られた返答と自身の経験的管理知識をもとに障害状況を判断し障害原因を絞り込む．

4. **調査情報の引き継ぎ**

自身の経験的管理知識による原因特定が完了すると，K-AIRから他のすべてのK-AIRに調査結果が通知される．これにより，1つのK-AIRが絞り込んだ障害原因をさらに詳細な障害原因を調査するK-AIRが処理を引き継いで原因特定を行い段階的に障害原因を特定する．

5. **調査結果の出力**

調査結果を引き継ぐK-AIRがいなくなった場合は，発生した障害に対してそのK-AIRの持つ経験的管理知識が最も詳細な原因とその対策を含んでいると判断することができる．この状態になると，K-AIRはAIR-NMSのインタフェースを通して自身のもつ経験的管理知識をネットワーク管理者に提示してAIR-NMSのネットワークサービス原因特定を終了する．経験的管理知識には特定された障害原因やその原因を取り除くためにネットワーク管理者が行うべき対策が記述されており，ネットワーク管理者はAIR-NMSの出力どおりに対策を行うことでネットワークサービスの障害管理が可能になる．

2.3　本章のまとめ

本章では，本研究の対象とするI-AIRを構成要素として持つ能動的情報資源の概念を用いたネットワーク管理支援システム（AIR-NMS）についてその詳細を述べた．またその中でネットワーク管理においてI-AIRの持つ情報資源用の情報の取得や保持が管理者の負担となっていることを述べた．次章では，さまざまなネットワークドメインに対応可能なI-AIRの詳細設計とその実装について述べる．

第3章　ネットワーク管理情報の利用を支援するAIRの提案

本章では，本研究の提案と様々なネットワークドメインに対応可能なI-AIRの設計の詳細を述べる．本章の構成として，はじめに本研究のアプローチを実現するために考慮すべき点とそれらを満たすために必要なI-AIRの機能要件を述べる．次にI-AIRを具体的に実現する方法としてエージェントフレームワークを用いた設計方法を述べ，I-AIRの各要素がエージェントフレームワークを用いてどのように実現されるかを述べる．

3.1　I-AIRの機能要件

本研究の目的である，管理者にやさしいNMSを実現するためには，以下の点を考慮する必要がある．

- 特定のドメイン環境において柔軟に情報取得できる設計
- 特有プラットホームの情報取得方法に対応した設計
- 取得情報を特定の環境によらず，保持できる設計

本研究では，これらを，自律的・能動的な情報取得支援および運用・再利用が容易な情報管理機能の実現によって，これらの点について考慮した設計をしている．以下では，この詳細を述べる．

3.2　自律的・能動的な情報取得支援

3.2.1　特定のドメイン環境において，柔軟に情報が取得できる設計

特定のプラットホームにおいて，情報を取得する場合，複数の取得方法が考えられる．例えば，WindowsXPにおいて情報を取得する場合，コマンドを使って情報を取得する方法が考えられるが，この場合，取得できない情報は他の方法を使って取得する必要がある．このため，本研究では，情報を柔軟，かつ，統一的に取得する為の情報取得方法として，情報取得定義ファイルを導入した．情報取得方法の定義ファイルは，オペレーティングシステムのプラットホームごとに用意してあり，情報取得項目ごとに取得方法が記載されている．情報取得プログラムでは，ネットワークドメインを判定した上で，該当するプラットホームのファイルを読み込むことによって，情報取得を行っている．ドメイン環境の判断は，Java言語で記述されている取得プログラム内のget.Propertyメソッドを用いて，OS名を取得することで判断している．情報取得ファイルに記述される項目は次の4項目である．

- 取得項目
- コマンドなどの具体的な取得方法
- 比較文字列
- 出力文字列

取得項目とは，I-AIRの情報資源に格納されるOSの名称などのように，取得すべき項目の名称のことである．また，比較文字列とは，取得方法を用いた結果，出力される文字列のことである．出力文字列とは，取得結果が比較文字列と同じ場合，出力する文字列のことである．情報取得定義ファイルの具体例を表3.1に示す．このファイルでは，”NIC DHCPEnabled”という項目を取得した結果「0」と出力された場合は，DHCPが無効であることを意味することから”Disable”と出力し，「1」が出力された場合は，DHCPが有

表 3.1: 情報取得定義ファイルの一例

取得項目	取得方法	比較文字列	出力文字列
NIC DHCPEnabled	WMI	0	Disable
NIC DHCPEnabled	WMI	-1	Enabled

効であることを意味することから”Enabled”と出力するということを意味している．このように，取得項目ごとに定義ファイル内に記述することによって，今後，現時点よりも多くの情報が取得できるようになった場合，定義ファイル内に，項目数を増やすことで対応することを可能にする．

3.2.2　特有プラットホームの情報取得方法に対応した設計

様々な環境の特性から，同様の内容の情報を異なる環境から取得する場合，それぞれの環境特性に応じた方法を用いる必要がある．情報の取得方法はプラットホームによって異なるため，多くの環境での情報取得方法について検討を重ねた結果，WindowsOSにおいては，SNMPよりも詳細な情報を取得可能なWMI（Windows Media Instrumentation）を，LinuxOSでは，ディストリビューションなどの違いのように，各違いを統一する為にSNMP[7]を用いることが妥当であるという結論に至った．このように，環境ごとに適している取得方法も異なるため，本研究では，環境に応じた情報取得プラグインを用いることでこの違いを吸収する．情報取得プラグインとは，ある環境上の情報取得を行う機能であり，環境ごとに，それぞれ特有の情報取得が可能な機能を設計することで，特定のプラットホームにしか存在しない取得方法の違いを吸収することを可能にしているものである．(図3.1))

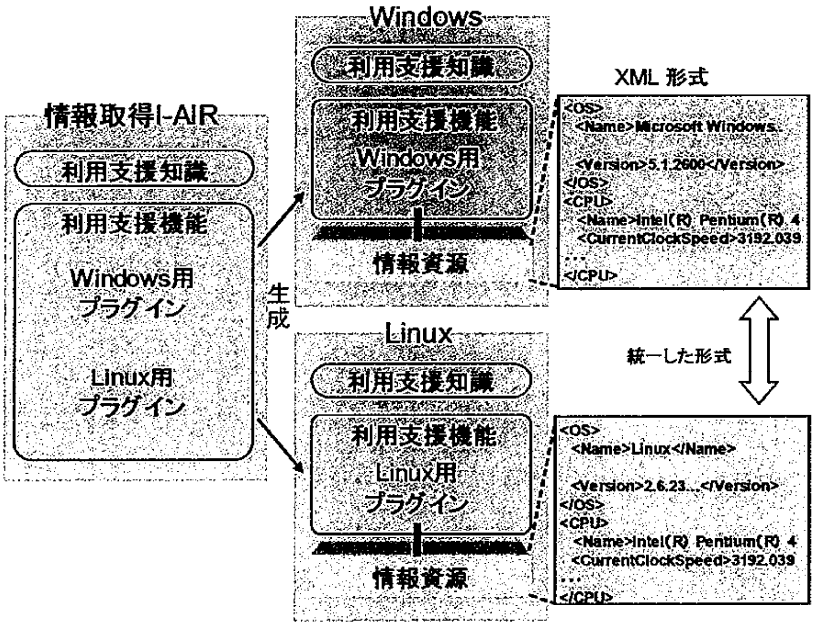

図 3.1: 環境に応じた情報取得の流れの組織図

3.3　運用・再利用が容易な情報管理機能の実現

3.3.1　取得情報を特定の環境によらず，保持できる設計

異なるネットワークドメイン環境から，取得した情報は，取得時に使ったコマンドやドメイン環境によって形式が異なる．そのため，原型のまま情報資源に格納した場合，情報資源がそのドメイン環境に依存した形式になってしまい，ドメイン環境に変更（設定など）があった場合，情報資源をすべて生成しなおす必要が出てくる．情報資源の記述形式を特定のネットワークドメイン環境によらない形で定める必要がある．本研究では，取得情報の項目ごとに，プラットホーム関係なく共通のタグを用意し取得し，情報をこのタグで囲み，XML/RDF形式で記述することで統一する．また，取得した情報は，タグで囲んだ上で，目的別に以下の3種類に分類する．具体的な分類項目は，図3.2のとおりである．

- トポロジ

  サブネットや接続機器の所在情報

分類項目	記述項目
トポロジー	IPAddress
ノード	OS (caption, CSDversion, Version, Buildnumber) CPU(Caption, Version, CurrentClockSpeed) MEMORY(TotalVisible MemorySize, Free Visible Memory Size, Total Virtual MemorySize, Free Virtual MemorySize) DISK(Caption, Size, FreeSpace) NIC(Caption)
サービス	Caption, Service Name, DHCPEnabled, IPaddress, DHS Hostname, DNS Server, MAC, IPSubnet, DefaultGateway

図 3.2: 取得情報の分類項目一覧

- サービス

  ネットワーク上の個々の端末の中身に関する詳細情報

- ノード

  ネットワークシステム上で動いているサービス（SNMP，メール等）に関する情報

このように分類して保持することで，管理者が直接情報資源を管理しやすくなるだけでなく，例えば，ネットワークドメイン全体のトポロジーやネットワークドメイン上のサービスについて知りたいとき，あるいは，特定の端末内部の情報のみを知りたい時に参照することが可能になるという利点がある．

3.4　詳細設計

3.4.1　状態遷移

本節では，提案手法を導入したI-AIRの各要素について述べる．I-AIRは基本的に待機状態になっており，外部からの要求を受信することで，各状態へ遷移し内部処理を行い，メッセージ待機状態へも度る設計になっている．各状態は次のとおりである．

- 待機状態

  外部から要求が来るのを待機している状態のことである．受信内容に基づき，この状態を起点としてほかの状態へ遷移して処理し，それ以外は待機状態になっている．本研究の設計では，情報取得要求を受けて，環境判断状態に遷移することになる．

- 環境判断状態

  情報を取得する対照の環境を判定する状態．環境の判断に失敗した場合は，待機状態に戻る．

- 情報取得状態

  環境判断状態での判断結果に基づき，実際に情報を取得する状態．情報取得定義ファイルに記述されている項目に基づき，取得可能項目すべてについて情報取得を行い，終了後は情報格納状態に遷移する．

- 情報格納状態

  情報取得状態において，取得した情報を分類し，保持する状態．終了後は待機状態に戻る．

3.4.2　情報資源

情報資源の記述形式としては，RDF/XML形式[5]を用いる．この形式を用いる利点として，タグを用いてメタ情報を記述することが可能であり，また，ツリー構造に対応した形式であることが挙げられる．このことから，ネットワークの情報を格納する際，格納した情報の構造を把握しやすくなるということがいえる．また，近年，NETCONF[11]に代表されるように，ネットワーク情報関連の規格は，RDF/XML形式で統一する傾向があり，将来的にも有効な表現法といえる．

3.4.3　利用支援知識

利用支援知識は，5つの知識から構成される．このうち，ID，IR，CPはAIRが動作する為に，基本的に持っている知識である．

- AIR識別知識 (AIR Identifier:ID)
- 情報資源に関する知識 (Information Resource: IR)
- 協調プロトコル (Communication Protocol: CP)
- 情報取得に関する知識 (Control Method:CM)
- 情報格納に関する知識 (Storage Method:SM)

3.4.4　利用支援機能

利用支援機能とは，ネットワークドメイン環境から情報を取得し，情報資源に格納しておく機能である．他のAIRから受けた要求に基づいて，どのような方法で情報を取得し，どのように情報資源に格納するかを定義した機能である．これらの機能を実現する為に次のような機能を用意しておくことが求められる．

- ネットワークドメイン環境を判断する機能
- 状態情報を取得する機能
- 情報を分類し格納する機能

3.5　本章のまとめ

本章では，管理者の負担を軽減し，管理者に優しいNMSを実現するために必要なI-AIRを実現するために必要な機能要件について述べ，次にその機能要件を満たすようなI-AIRの各要素の詳細設計を述べた．次の章では，本章で提案した設計の具体的な実装および評価実験について述べる．

第4章　実装と評価実験

本章では，本研究で提案した手法の実装および評価実験について述べる．本章の構成として，はじめに第3章の設計に基づいた実装について述べ，次に提案手法の有効性を評価する為の評価実験について述べる．

4.1　実装

前章で述べたとおり，I-AIRの各要素（状態遷移，情報資源，利用支援知識，利用支援機能，協調プロトコル）を設計することで，管理者がNMSを利用する際の管理負担を軽減することができる．このような形のAIRは，ルールベースの知識に基づき活動するプログラムとして実装される．次の節では，その実装方法を述べる．

4.1.1　エージェントによる実現方法

各AIRは，分散環境上にある情報資源に利用支援知識と利用支援機能というエージェントとしての機能を付加することで実現される．このようにすることで，各AIRの情報資源は能動的・自律的に動作することが可能になり，情報資源に対する作業の一部を代行することが可能になる．このようなAIRの実現方法としてマルチエージェントを用いる方法が提案されている [8]，[9]．この方法では，各AIRの持つ，次のような特徴を実現する上で，マルチエージェントシステムが提供する機能や動作特性が効果的に利用できることによるものである．

- 外部からの要求に応じて，処理を行う
- 複数のAIRが協調・連携することで問題を解決する

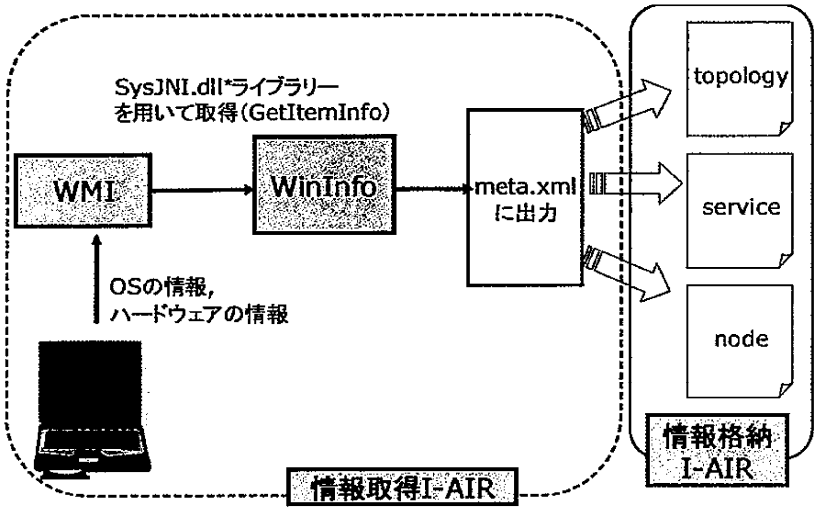

図 4.1: 実装したプログラムの実際の流れ

- 利用支援知識に基づいて活動を行う

4.1.2　実装環境

- OS: Windows XP Professional, Ubunto 8.0.4
- プログラミング言語（ベースプロセス）：JAVA 言語（J2SE 1.6.0）
- IDEA: 1.3.1

実装したプログラムの実際の流れは，図4.1 のようになっている．なお，図中の WinInfo[10] プログラムとは，Windows のダイナミックライブラリのひとつである WMI(Windows Management Instrumentation) を用いて情報を取得するプログラムである．

4.2　評価実験

本研究の最終目的は，NMS の利用負担を軽減し，管理者にやさしい NMS を実現することである．ネットワーク管理において，負担とは，管理者が作業を行う際に感じるものを指す．具体的には，管理に要する作業量，つまり，使用したコマンドやプロセスの数な

どに比例する．よって，本研究で設計したI-AIRを使用前と使用後の使用したコマンドやプロセスの数の比較をすることを通して，提案手法の有効性を調べ，評価を行う．

[実験1]

この実験では，異なるネットワークプラットホーム下における提案システムの有効性を評価する．具体的には，以下のような環境で提案するシステムを導入し，システムが情報を取得する場合と，被験者が手動で情報を取得する場合の取得項目数について，それぞれ比較した．また，取得した情報は，I-AIRの場合と比較を容易にする為，表4.1を被験者に与え，これに沿った形でXMLを生成してもらった．

この実験の被験者は，コンピュータに関する知識はあるが，ネットワーク管理経験，NMS使用経験はない．

- WindowsOS

  WindowsXP　Profesional，Service Pack 3，CPU 1.6Ghz

- Unix系OS

  Ubuntu 8.04

[実験2]

この実験では，管理者のスキルの違いによる提案システムの有効性を評価する．具体的には，同一のプラットホーム上で，コンピュータの知識やネットワークの管理経験の異なる被験者に協力していただき，被験者が手動で情報を取得する場合の取得項目数について，それぞれ比較した．被験者1は，実験1と同じである．また，被験者2については，コンピュータに関する知識はほとんどない初心者である．

実験結果と考察を次の節で示す．

表 4.1: 情報取得項目

分類項目	記述項目
トポロジー	IPAddress
ノード	OS (caption, CSDversion, Version, Buildnumber) CPU(Caption, Version, CurrentClockSpeed) MEMORY(TotalVisible MemorySize, Free Visible Memory Size, Total Virtual MemorySize, Free Virtual MemorySize) DISK(Caption, Size, FreeSpace) NIC(Caption)
サービス	Caption, Service Name, DHCPEnabled, IPaddress, DHS Hostname, DNS Server, MAC, IPSubnet, DefaultGateway

4.3　実験結果

[実験 1]

実験 1 の具体的な結果を付録 1（被験者，WindowsOS），付録 2（I-AIR の取得情報，WindowsOS），付録 3（被験者，Unix 系 OS）および付録 4（I-AIR の取得情報，Unix 系 OS）に示す．この結果を分析したものを表 4.2，および，表 4.3 に示す．

[実験 2]

実験 2 の具体的な取得結果について，被験者 1 に関しては，実験 1 の被験者取得情報と同一であるため，省略する．また，被験者 2 については，付録 5（被験者の取得情報）に実験結果を示す．この結果を分析したものを表 4.4，および，表 4.5 に示す．

表 4.2: 実験 1 におけるプラットホーム別取得項目

実験1	取得項目の種類	Windows		Linux	
		システム	被験者	システム	被験者
OS	Caption	○	○	○	○
	CSDVersion	○			
	Version	○	○	○	
	BuildNumber	○			
CPU	Caption	○	○	○	○
	Version	○			
	CurrentClockSpeed	○	○	○	
MEMORY	TotalVisible MemorySize	○	○	○	○
	FreePhysical Memory	○	○	○	
	TotalVirtual MemorySize	○			
	FreeVirtual MemorySize	○			
DISK	Caption			○	
	Size			○	
	FreeSpace			○	
NIC	Caption	○	○	○	○
	ServiceName	○			
	DHCPEnabled	○			
	IPAddress	○	○	○	○
	DHSHostName				
	DHSServer SearchOrder				
	MACAddress	○	○	○	○
	IPSubnet	○	○	○	○
	DeffaultIPGateway	○	○	○	○

表 4.3: 実験 1 の取得項目数の合計

取得項目	システム（Win）	被験者（Win）	システム (Unix)	被験者（Unix）
合計	49	26	14	8

4.3.1　実験考察

実験 1 および 2 の結果から，次のことが言える．

- ネットワークプラットホームに関係なく被験者と I-AIR において取得した情報量に違いが見られる．

  例えば，被験者の場合，WindowsOS 内のマイコンピュータのプロパティを参照することでわかる，OS の正確な名称やバージョン，CPU の情報のように比較的に容易に取得できる情報については，I-AIR と同様に取得できていることがわかる．一方で，NIC の情報など専門的な知識が必要になってくる情報については，必要なテンプレートの数を把握することができず 1 種類しか取得できていない．また，Unix 系 OS である Linux の場合についても，プラットホームの名称や IP アドレスのように比較的取得方法が容易なものに関しては，被験者の場合も取得できているが，それ以外の情報に関しては取得できていないことがわかる．このことから，提案手法は取得方法が困難な情報についても，他の方法と同じように取得できることがわかる．

- 被験者のスキルのレベルの違いによる取得情報の違い

  実験 2 の結果から，異なる被験者で同様の実験を行った場合についても，専門的な知識が必要とされる情報については取得するにはいたらず，被験者のスキルが取得情報および取得項目数に影響に影響を及ぼしていることがわかる．

- 取得手法の違い

  被験者は，自身の知識に基づき，情報を収集していたが，提案システムの場合，最適な方法を判断した上でシステム内に組み込んでいるの．このため，複数の管理者

表 4.4: 実験2におけるプラットホーム別取得項目

実験2	取得情報の種類	システム	被験者1	被験者2
OS	Caption	○	○	○
	CSDVersion	○		
	Version	○	○	
	BuildNumber	○		
CPU	Caption	○	○	○
	Version	○		
	CurrentClockSpeed	○	○	
MEMORY	TotalVisibleMemorySize	○	○	○
	FreePhysicalMemory	○	○	
	TotalVirtualMemorySize	○		
	FreeVirtualMemorySize	○		
NIC	Caption	○	○	○
	ServiceName	○		
	DHCPEnabled	○		
	IPAddress	○	○	○
	DHSHostName			
	DHSServerSearchOrder			
	MACAddress	○	○	○
	IPSubnet	○	○	○
	DeffaultIPGateway	○	○	○

表 4.5: 実験2の取得項目数の合計

取得項目	システム	被験者1	被験者2
合計	49	26	7

がいる場合でも，統一した情報が取れる．

- 取得にかかった時間の短縮

  被験者が情報取得を行う場合，平均で１０数分かかるが，提案システムは，数秒から１分程度で情報取得できる．

以上のことから，管理者が手動で情報取得を行った場合，管理者のスキルによる情報の取得量に差はあるが，専門的な知識が必要な情報については，一般的に取得することが困難であることがいえる．また，専門的な知識がないために，取得情報の必要性の認識が不測しており，NICの数などは機器特有のものを認識できないなかったり，数などの，必要分のフォーマットを用意することができないことがわかる．

以上のことから，本研究で提案した手法を導入したことによって，管理者のスキルの差に関係なく，同様の内容の情報取得が可能であることがわかる．また，情報取得の際の手間が省け，情報を網羅的に取得することが可能になるため，管理スキルの高い管理者に対してもより多角的な判断を可能にする．このことから，提案手法を導入したことによって，機器固有の情報を適切に取得，判別できることから，NMSを利用する際の管理者の作業負担を軽減することが出来るといえる．

4.4　本章のまとめ

本章では，実際に実装してある環境についての説明を述べた．また，この環境下で実際に実験を行い，その結果を基に本研究で提案する手法の評価し，その有効性について考察した．

第5章　結論

第5章では，本研究の結論を述べる．

5.1　本論文のまとめ

ネットワークサービス障害管理の現状として，既存手法の解析ツールやNMSを用いた管理方法では，ネットワークドメインごとに対応した情報資源を用意し，管理を行うことが困難なため，最終的な情報取得から内容分析，格納までネットワーク管理者が行わなければならず，管理の負担となっていた．

その問題を解決する手法として，能動的情報資源の概念を用いたネットワーク管理支援システムAIR-NMSが提案されている．AIR-NMSはネットワークドメイン内の環境を考慮した経験的管理知識を管理支援に用いることで，既存手法を用いた管理手法の問題点を解決し，ネットワークサービス管理時に管理者にかかる負担を軽減することが可能である．しかし，現状のAIR-NMSを異なるネットワークドメインに適用しようとした場合，情報の取得方法や格納方法がネットワークドメインごとに異なっており，また，取得した情報をドメイン依存ごとに表現形式が異なっていたため，そのままの記述内容で適用させることが困難であった．これは現状のAIR-NMSに以下のような課題が残っているためである．

- ネットワークの知識があまりない人は，必要な情報を取得できない
- ネットワークドメイン環境の変化（機器の設定環境の変化，ネットワーク環境の変化）に自動的に対応することが困難

本稿では，AIR-NMSの利便性を高めるために，汎用的なI-AIRの情報資源の情報取得・

格納機能を提案した．この手法では，静的にAIR-NMS内に記述していた情報資源について，情報取得から状態情報として保持することにいたるまでの一連の流れを動的なものにすることを提案している．これによって情報取得や情報資源の新規作成するによるネットワーク管理者への負担の軽減が見込める．また，状態情報の更新にも動的に対応することが可能になる．

5.2　今後の課題

本研究では，ネットワークの知識があまりない人でも，NMSを利用する際に必要となる情報の取得を支援し，ネットワークドメイン環境に変化があった場合でも，自動的に必要な状態情報を取得し格納する形式を定義することによって，管理者にやさしいNMSの実現を目指した．提案手法を用いたI-AIRでは，自動的にドメイン環境を判断した上で，情報を取得し，保持することが可能である．

今後，提案手法をより発展させていく為に，現状では取れない情報の取得手法の確立や，取得じょうほを元に推論によって管理者に情報を提示する機能，またアクセス権限等を用いることで，各端末の内部の詳細な情報が他の端末からも見れる仕組みの確立などが求められてくる．

謝辞

本論文を終えるにあたり，本研究を行う機会を与えていただき，また研究方針について数々の貴重な御意見・御指導を頂いた東北大学サイバーサイエンスセンター，木下哲男教授に謹んで感謝致します.

また，本論文の審査をしていただき，有用なコメントをいただきました東北大学サイバーサイエンスセンター，曽根秀昭教授，並びに，東北大学大学院情報科学研究科，加藤寧教授に深く感謝致します.

本研究を進めるにあたり，日頃から有意義な御指導と御助言を下さいました，東北大学情報サイバーサイエンスセンター，阿部亨准教授，やわらかい情報システム研究センター，北形元准教授に深く感謝致します.

また，様々な場面において，私の研究へのアドバイスをくださった千葉工業大学工学部准教授，今野将博士，東北大学電気通信研究所助教，矢入聡博士，東北大学電気通信研究所教育研究支援者，笹井一人博士に深く感謝いたします.

最後に，研究活動において貴重なご意見・助言を頂きました木下研究室の皆様に心から感謝します.

参考文献

[1] Alan Bivens et.al. "Scalability and Performance of an agent-based network management middleware", International Journal of Network Management, pp131-146, 2004.

[2] 木下哲男, "分散情報資源活用の一手法　—能動的情報資源の設計—", 信学技報, AI99-45, pp.13-19, 1999.

[3] Susumu Konno, Yukio Iwaya, Toru Abe, Tetsuo Kinoshita, "Knowledge-based Support of Network Management Tasks Using Active Information Resource", IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp.195-199, 2006.

[4] 吉村智志, "ネットワークシステムの自律的な状態監視機構に関する研究", 平成16年度　東北大学情報科学研究科修士学位論文

[5] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, Francois Yergeau eds, "Extensible Markup Language (XML) 1.0(Fourth Edition)", W3C Recommendation 16 August 2006.

[6] DASH - Distributed Agent System based on Hybrid architecture, http://www.agent-town.com/dash/

[7] J.Case, K.McCloghrie, M, Rose, S.Waldbusser, "Introduction to Community-based SNMPv2", RFC1901, April1993

[8] Baoning Li, Toru Abe, Kenij Sugawara, Tetsuo Kinoshita, "Active Information Resource: Design concept and example", Proceedings of 17th International Conference on Advanced Information Networking and Applications(AINA2003), pp.274-277, 2003.

[9] 今野将, 吉村智志, 岩谷幸雄, 阿部亨, 木下哲男, "能動的情報資源を用いた自律的なネットワーク監視システム", 第4回科学技術フォーラム FIT2005 情報技術レターズ4 LF-009, pp.111-114, 電気情報通信学会/情報処理学会, 2005.

[10] 駒形順子, "Windows 情報取得についての調査報告", 2006

[11] IETF Network Working Group, "NETCONF Configuration Protocol", RFC4741, 2006.

研究発表論文

発表論文一覧

1. バーラ イシャ，今野 将，木下 哲男，"AIR-NMSにおける状態情報の記述法の検討"，電子情報通信学会2008年総合大会講演論文集，D-8-10, 2008.

2. 前田　哲，バーラ　イシャ，今野　将，木下　哲男，"AIR-NMSのための端末の状態情報取得手法"，電子情報通信学会2008年総合大会講演論文集，D-8-9, 2008.

3. 前田　哲，バーラ　イシャ，今野　将，木下　哲男，"AIR-NMSにおける異種プラットフォームへの適応性を考慮したネットワーク機器情報取得手法"，平成20年度電気関係学会東北支部連合大会講演論文集，2F07, 2008.

4. バーラ イシャ, 前田　哲，今野 将, 木下 哲男, "AIR-NMSにおける汎用的I-AIRの設計", FIT2008(第7回情報科学技術フォーラム) 講演論文集, F-005, 2008.

付録

付録1

[実験1の実験結果]

(被験者取得情報，Windows)

被験者が取得した情報は次のとおりである．

Windows IP Configuration

Host Name . . . . . : kinolab-78b356c

Primary Dns Suffix . . . . . :

Node Type . . . . . : Unknown

IP Routing Enabled. . . . . : No

WINS Proxy Enabled. . . . . : No

Ethernet adapter ワイヤレス ネットワーク接続:

Media State . . . . . : Media disconnected

Description . . . . . : Intel(R) PRO/Wireless 3945ABG Network Connection

Physical Address. . . . . : 00-1C-BF-4E-01-26

Ethernet adapter ローカル エリア接続:

Connection-specific DNS Suffix . :

Description . . . . . : Realtek RTL8139/810x Family Fast Ethernet NIC

Physical Address. . . . . : 00-0B-97-50-B2-63

Dhcp Enabled. . . . . : Yes

Autoconfiguration Enabled . . . . : Yes

IP Address. . . . . : 172.20.1.45

Subnet Mask . . . . . : 255.255.255.0

Default Gateway . . . . . : 172.20.1.1

DHCP Server . . . . . : 172.20.1.1

DNS Servers . . . . . : 172.20.1.19, 130.34.208.194

Lease Obtained. . . . . : 2009 年 2 月 12 日 11:43:03

Lease Expires . . . . . : 2009 年 2 月 13 日 11:43:03

システム:

Microsoft WindowsXP Professional

Version 2002

Service Pack 3

User: Bhalla

コンピュータ:

Intel(R) Core(TM)2 CPU

U7500 @1.06GHz

1.06 GHz, 1.49GB RAM

物理メモリ (KB):

合計:1562604

利用可能:834008

また，この結果をもとに生成させた XML 形式を次のページに示す.

付録2

[実験1の実験結果]

(J-AIR の取得情報，Windows)

```
<?xml version="1.0" encoding="Shift_JIS" ?>
- <SUBNET>
    <name />
  - <USER>
      <name />
    - <OS>
        <Caption>Microsoft Windows XP Professional</Caption>
        <CSDVersion>Service Pack 3</CSDVersion>
        <Version />
        <BuildNumber />
      </OS>
    - <CPU>
        <Caption>Intel(R)  Core(TM)2  CPU</Caption>
        <Version />
        <CurrentClockSpeed>1.06 GHz</CurrentClockSpeed>
      </CPU>
    - <MEMORY>
        <TotalVisibleMemorySize>1562604</TotalVisibleMemorySize>
        <FreePhysicalMemory>834008</FreePhysicalMemory>
        <TotalVirtualMemorySize />
        <FreeVirtualMemory />
      </MEMORY>
    - <NIC>
        <Caption>Realtek RTL8139/810x Family Fast Ethernet NIC</Caption>
        <ServiceName />
        <DHCPEnabled />
        <IPAddress>172.20.1.45</IPAddress>
        <DHSHostName />
        <DNSServerSeachOrder />
        <MACAddress>00-0B-97-50-B2-63</MACAddress>
        <IPSubnet>255.255.255.0</IPSubnet>
        <DefaultIPGateway>172.20.1.1</DefaultIPGateway>
      </NIC>
  </USER>
</SUBNET>
```

付録3

[実験1の実験結果]

（被験者の取得情報，Unix）

```
<?xml version="1.0" encoding="Shift_JIS" standalone="no" ?>
- <SUBNET>
  - <USER>
    - <OS>
        <Caption>Linux</Caption>
        <CSDVersion />
        <Version />
        <BuildNumber />
      </OS>
    - <CPU>
        <Caption>Intel(R) Pentium(R) 4 CPU 3.20GHz</Caption>
        <Version />
        <CurrentClockSpeed />
      </CPU>
    - <MEMORY>
        <TotalVisibleMemorySize>1034400 kB</TotalVisibleMemorySize>
        <FreePhysicalMemory />
      </MEMORY>
    - <DISK>
        <Caption />
        <Size />
        <FreeSpace />
      </DISK>
    - <NIC>
        <Caption>Ethernet</Caption>
        <ServiceName />
        <DHCPEnabled />
        <IPAddress>172.20.1.7</IPAddress>
        <DHSHostName />
        <DNSServerSearchOrder />
        <MACAddress>00:0C:F1:93:60:20</MACAddress>
        <IPSubnet>255.255.255.0</IPSubnet>
        <DefaultIPGateway>172.20.1.255</DefaultIPGateway>
      </NIC>
  </USER>
</SUBNET>
```

付録4

[実験1の実験結果]

(I-AIRの取得情報，Unix)

```
<?xml version="1.0" encoding="Shift_JIS" standalone="no" ?>
- <SUBNET>
  - <USER>
    - <OS>
        <Caption>Linux</Caption>
        <CSDVersion />
        <Version>2.6.23.1-42.fc8</Version>
        <BuildNumber />
      </OS>
    - <CPU>
        <Caption>Intel(R) Pentium(R) 4 CPU 3.20GHz</Caption>
        <Version />
        <CurrentClockSpeed>3192.196</CurrentClockSpeed>
      </CPU>
    - <MEMORY>
        <TotalVisibleMemorySize>1034400 kB</TotalVisibleMemorySize>
        <FreePhysicalMemory>15028 kB</FreePhysicalMemory>
      </MEMORY>
    - <DISK>
        <Caption>/dev/sda3</Caption>
        <Size>194449</Size>
        <FreeSpace>69592752</FreeSpace>
      </DISK>
    - <NIC>
        <Caption>Ethernet</Caption>
        <ServiceName />
        <DHCPEnabled />
        <IPAddress>172.20.1.7</IPAddress>
        <DHSHostName />
        <DNSServerSearchOrder />
        <MACAddress>00:0C:F1:93:60:20</MACAddress>
        <IPSubnet>255.255.255.0</IPSubnet>
        <DefaultIPGateway>172.20.1.255</DefaultIPGateway>
      </NIC>
    </USER>
  </SUBNET>
```

付録5

[実験2の実験結果]

（被験者2の場合，被験者1の結果は実験1のWindowsの結果と同じ）

```
<?xml version="1.0" encoding="Shift_JIS" standalone="no" ?>
- <SUBNET>
    <name />
  - <USER>
      <name />
    - <OS>
        <Caption>Microsoft Windows XP Professional</Caption>
        <CSDVersion>Service Pack 2</CSDVersion>
        <Version>5.1.2600</Version>
        <BuildNumber />
      </OS>
    - <CPU>
        <Caption>Intel(R) Core(TM)2 CPU</Caption>
        <Version />
        <CurrentClockSpeed>1.06 GHz</CurrentClockSpeed>
      </CPU>
    - <MEMORY>
        <TotalVisibleMemorySize>1047548</TotalVisibleMemorySize>
        <FreePhysicalMemory>361040</FreePhysicalMemory>
        <TotalVirtualMemorySize />
        <FreeVirtualMemory />
      </MEMORY>
    </USER>
  </SUBNET>
```